WHAT IS CLAIMED IS:

1. A graphics processing subsystem for a computer system having a global address space, the graphics processing subsystem comprising:

a plurality of memories including a first memory and a second memory, each of the plurality of memories including a plurality of addressable storage locations, wherein a first storage location in the first memory and a second storage location in the second memory are addressable by a common global address in the global address space; and

a plurality of graphics processors including a first graphics processor associated with the first memory,

wherein the first storage location in the first memory and the second storage location in the second memory are uniquely identifiable by respective first and second private addresses internal to the graphics processing subsystem, and

wherein the first graphics processor is configured to access the second storage location in the second memory by referencing the second private address.

2. The graphics processing subsystem of claim 1, wherein the first graphics processor is further configured to access the first storage location in the first memory by referencing the common global address.

3. The graphics processing subsystem of claim 1, wherein the first graphics processor is further configured to access the first storage location in the first memory by referencing a local address.

4. The graphics processing subsystem of claim 3, wherein the local address includes an offset value.

5. The graphics processing subsystem of claim 1, further comprising:

a bridge unit coupled to the plurality of memories and configured to convert private addresses of the storage locations to global addresses.

6. The graphics processing subsystem of claim 5, wherein the bridge unit is further configured to communicate with a system bus of the computer system.

7. The graphics processing subsystem of claim 6, wherein the bridge unit is further configured to receive a memory access request that originates from one of the

plurality of graphics processors and references a global address that does not correspond to any of the storage locations in the plurality of memories and to respond to the memory access request by communicating with the system bus.

8. The graphics processing subsystem of claim 6, wherein the bridge unit is further configured to receive a memory access request via the system bus, the memory access request referencing the common global address, and to broadcast the memory access request to the first memory and the second memory.

9. The graphics processing subsystem of claim 5, further comprising:

a first memory interface coupled between the first graphics processor and the first memory and configured to receive a memory access request from the first graphics processor.

wherein the first memory interface is further configured to respond to the memory access request by accessing the first memory in the event that the memory access request references the common global address.

10. The graphics processing subsystem of claim 9, wherein the first memory interface is further configured to forward the memory access request to the bridge unit in the event that the memory access request references an address other than an address of one of the storage locations in the first memory.

11. The graphics processing subsystem of claim 5, wherein the plurality of graphics processors further includes a second graphics processor associated with the second memory,

wherein the second graphics processor is configured to access the first storage location in the first memory by referencing the first private address.

12. The graphics processing subsystem of claim 11, wherein the second graphics processor is further configured to access the first storage location in the second memory by referencing the common global address.

13. The graphics processing subsystem of claim 11, wherein the second graphics processor is further configured to access the first storage location in the second memory by referencing a local address.

14. The graphics processing subsystem of claim 13, wherein the local address includes an offset value.

15. A graphics processing subsystem for a computer system having a global address space, the graphics processing subsystem comprising:

a first memory and a second memory, each including a plurality of addressable storage locations;

a first memory interface and a second memory interface coupled to the first memory and the second memory, respectively;

a first graphics processor and a second graphics processor coupled to the first memory interface and the second memory interface, respectively; and

a bridge unit coupled to each of the first and second memory interfaces and configured to assign a unique private address to each of the storage locations in each of the first and second memories,

wherein the first memory interface is configured to receive a first memory access request including an address from the first graphics processor, to respond to the first memory access request by accessing the first memory in the event that the address matches an address of a storage location in the first memory, and to forward the first memory access request to the bridge processor in the event that the address does not match an address of a storage location in the first memory, and

wherein the bridge unit is configured to process the first memory access request by accessing the second memory interface in the event that the address matches a private address of one of the storage locations in the second memory.

16. The graphics processing subsystem of claim 15, wherein the bridge unit is further configured to process the first memory access request by accessing a system bus of the computer system in the event that the address does not match a private address of one of the storage locations in the second memory.

17. The graphics processing subsystem of claim 15, wherein the second memory interface is configured to receive a second memory access request including an address from the second graphics processor, to respond to the second memory access request by accessing the second memory in the event that the address matches an address of a storage location in the second memory, and to forward the second memory access request to the

bridge processor in the event that the address does not match an address of a storage location in the second memory.

18. The graphics processing subsystem of claim 17, wherein the bridge unit is configured to process the second memory access request by accessing the first memory interface in the event that the address matches a private address of one of the storage locations in the first memory.

19. A method for accessing data in a graphics processing subsystem that has a first graphics processor and a second graphics processor coupled to a first memory and a second memory, respectively, the method comprising:

assigning a global address to each of a plurality of storage locations in the first memory and to each of a plurality of storage locations in the second memory, wherein a first storage location in the first memory and a second storage location in the second memory are assigned a common global address;

assigning a respective unique private address to each of the storage locations in each of the first and second memories, wherein the private addresses are internal to the graphics processing subsystem;

receiving a first memory access request originating from the first graphics processor and referencing a target address that matches a private address of one of the storage locations in the second memory;

identifying a target storage location in the second memory based on the matched private address; and

accessing the target storage location in the second memory.

20. The method of claim 19, further comprising:

receiving a second memory access request originating from the second graphics processor and referencing a target address that matches a private address of one of the storage locations in the first memory;

identifying a target storage location in the first memory based on the matched private address; and

accessing the target storage location in the first memory.

21. The method of claim 19, further comprising:

receiving a second memory access request originating from the first graphics processor and referencing the common global address; and

responding to the second memory access request by accessing the first storage location in the first memory.

22. The method of claim 21, further comprising:

receiving a third memory access request originating from the second graphics processor and referencing the common global address; and

responding to the third memory access request by accessing the second storage location in the second memory.

23. The method of claim 19, further comprising:

receiving a second memory access request that originates from one of the first and second graphics processors and references a global address that does not correspond to any of the storage locations in the first and second memories; and

responding to the memory access request by accessing the system bus.

24. The method of claim 19, further comprising:

receiving a second memory access request via the system bus, the memory access request referencing the common global address; and

broadcasting the second memory access request to the first memory and the second memory.

25. The method of claim 19, wherein the identified storage location is protected by a semaphore, the method further comprising:

acquiring the semaphore prior to accessing the identified storage location; and

releasing the semaphore after accessing the identified storage location,

wherein after acquiring the semaphore and before releasing the semaphore, access to the identified storage location by the second graphics processor is not permitted.

26. The method of claim 25, wherein the semaphore is stored in a semaphore storage location in the second memory.

27. The method of claim 25, wherein the semaphore is stored in a semaphore storage location in the first memory.

28. A method for accessing data in a graphics processing subsystem that has a first graphics processor and a second graphics processor coupled to a first memory and a second memory, respectively, the method comprising:

assigning a global address to each of a plurality of storage locations in the first memory and to each of a plurality of storage locations in the second memory, wherein a first storage location in the first memory and a second storage location in the second memory are assigned a common global address;

assigning a unique private address to each of the storage locations in each of the first and second memories, wherein the private addresses are internal to the graphics processing subsystem;

receiving a data transfer request originating from the first graphics processor, the data transfer request referencing a source address that matches the common global address and referencing a destination address that matches a private address of one of the storage locations in the second memory;

identifying as a source location the first storage location in the first memory;

identifying a destination location in the second memory based on the matched private address; and

initiating a data transfer from the source location to the destination location.

29. A method for accessing data in a graphics processing subsystem that has a first graphics processor and a second graphics processor coupled to a first memory and a second memory, respectively, the method comprising:

assigning a global address to each of a plurality of storage locations in the first memory and to each of a plurality of storage locations in the second memory, wherein a first storage location in the first memory and a second storage location in the second memory are assigned a common global address;

assigning a unique private address to each of the storage locations in each of the first and second memories, wherein the private addresses are internal to the graphics processing subsystem;

receiving a data transfer request originating from the first graphics processor, the data transfer request referencing a source address that matches a private address of one of the storage locations in the second memory and referencing a destination address that matches the common global address;

identifying a source storage location in the second memory based on the matched private address;

identifying as a destination location the first storage location in the first memory; and

initiating a data transfer from the source location to the destination location.